Package 'dtmapi'

May 8, 2026

Title Fetching Data from the 'Displacement Tracking Matrix'

Version 0.1.0

Description Allows humanitarian community, academia, media, government, and non-governmental organizations to utilize the data collected by the 'Displacement Tracking Matrix' (<https://dtm.iom.int>), a unit in the International Organization for Migration. This also provides non-sensitive Internally Displaced Person figures, aggregated at the country, Admin 1 (states, provinces, or equivalent), and Admin 2 (smaller administrative areas) levels.

URL https://github.com/Displacement-Tracking-Matrix/dtmapi-R, https://displacement-tracking-matrix.github.io/dtmapi-R/

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports httr2, testthat (>= 3.1.0), askpass

Suggests knitr, rmarkdown, withr

VignetteBuilder knitr

BugReports https://github.com/Displacement-Tracking-Matrix/dtmapi-R/issues

NeedsCompilation no

Author Luong Bang Tran [aut],
Assad Asil Companioni [aut, cre],
Displacement Tracking Matrix [cph]

Maintainer Assad Asil Companioni <aasil@iom.int>

Repository CRAN

Date/Publication 2025-08-23 22:00:21 UTC

Contents

`get_all_countries` *Fetch All Countries*

Description

Retrieve all countries for which DTM data is publicly available through the API.

Usage

```
get_all_countries()
```

Value

A data frame containing the list of all countries.

Examples

```
countries_df <- get_all_countries()
head(countries_df)
```

`get_all_operations` *Fetch All Operations*

Description

Retrieve all operations for which DTM data is publicly available through the API.

Usage

```
get_all_operations()
```

Value

A data frame containing the list of all operations.

Examples

```
# Fetch all operations
operations_df <- get_all_operations()
head(operations_df)
```

`get_idp_admin0_data` *Fetch IDP Admin0 Data*

Description

Retrieve IDP data at Admin 0 level based on specified parameters. At least one of the following parameters must be provided: Operation, CountryName, or Admin0Pcode.

Usage

```
get_idp_admin0_data(
  Operation = NULL,
  CountryName = NULL,
  Admin0Pcode = NULL,
  FromReportingDate = NULL,
  ToReportingDate = NULL,
  FromRoundNumber = 0,
  ToRoundNumber = 0
)
```

Arguments

`Operation`	Optional; Name of the DTM operation for which the data was collected.
`CountryName`	Optional; Name of the country where the data was collected.
`Admin0Pcode`	Optional; Country code (ISO 3166-1 alpha-3).
`FromReportingDate`	Optional; Start date for the reporting period (format: 'YYYY-MM-DD').
`ToReportingDate`	Optional; End date for the reporting period (format: 'YYYY-MM-DD').
`FromRoundNumber`	Optional; Starting round number for the data collection range.
`ToRoundNumber`	Optional; Ending round number for the data collection range.

Value

A data frame containing the IDP Admin0 data matching the specified criteria.

Examples

```
# Fetch IDP data at Admin Level 0
idp_admin0_df <- get_idp_admin0_data(CountryName = "Ethiopia",
                                     FromRoundNumber = 1,
                                     ToRoundNumber = 10)
head(idp_admin0_df)
```

`get_idp_admin1_data` *Fetch IDP Admin1 Data*

Description

Retrieve IDP data at Admin 1 level based on specified parameters. At least one of the following parameters must be provided: Operation, CountryName, or Admin0Pcode.

Usage

```
get_idp_admin1_data(
  Operation = NULL,
  CountryName = NULL,
  Admin0Pcode = NULL,
  Admin1Name = NULL,
  Admin1Pcode = NULL,
  FromReportingDate = NULL,
  ToReportingDate = NULL,
  FromRoundNumber = 0,
  ToRoundNumber = 0
)
```

Arguments

`Operation`	Optional; Name of the DTM operation for which the data was collected.
`CountryName`	Optional; Name of the country where the data was collected.
`Admin0Pcode`	Optional; Country code (ISO 3166-1 alpha-3).
`Admin1Name`	Optional; Name of level 1 administrative boundaries.
`Admin1Pcode`	Optional; Place code of level 1 administrative boundaries.
`FromReportingDate`	Optional; Start date for the reporting period (format: 'YYYY-MM-DD').
`ToReportingDate`	Optional; End date for the reporting period (format: 'YYYY-MM-DD').
`FromRoundNumber`	Optional; Starting round number for the data collection range.
`ToRoundNumber`	Optional; Ending round number for the data collection range.

Value

A data frame containing the IDP Admin1 data matching the specified criteria.

Examples

```
# Fetch IDP data at Admin Level 1
idp_admin1_df <- get_idp_admin1_data(CountryName = "Sudan", Admin1Name = "Blue Nile")
head(idp_admin1_df)
```

`get_idp_admin2_data` *Fetch IDP Admin2 Data*

Description

Retrieve IDP data at Admin 2 level based on specified parameters. At least one of the following parameters must be provided: Operation, CountryName, or Admin0Pcode.

Usage

```
get_idp_admin2_data(
  Operation = NULL,
  CountryName = NULL,
  Admin0Pcode = NULL,
  Admin1Name = NULL,
  Admin1Pcode = NULL,
  Admin2Name = NULL,
  Admin2Pcode = NULL,
  FromReportingDate = NULL,
  ToReportingDate = NULL,
  FromRoundNumber = 0,
  ToRoundNumber = 0
)
```

Arguments

`Operation`	Optional; Name of the DTM operation for which the data was collected.
`CountryName`	Optional; Name of the country where the data was collected.
`Admin0Pcode`	Optional; Country code (ISO 3166-1 alpha-3).
`Admin1Name`	Optional; Name of level 1 administrative boundaries.
`Admin1Pcode`	Optional; Place code of level 1 administrative boundaries.
`Admin2Name`	Optional; Name of level 2 administrative boundaries.
`Admin2Pcode`	Optional; Place code of level 2 administrative boundaries.
`FromReportingDate`	Optional; Start date for the reporting period (format: 'YYYY-MM-DD').
`ToReportingDate`	Optional; End date for the reporting period (format: 'YYYY-MM-DD').
`FromRoundNumber`	Optional; Starting round number for the data collection range.
`ToRoundNumber`	Optional; Ending round number for the data collection range.

Value

A data frame containing the IDP Admin2 data matching the specified criteria.

Examples

```
# Fetch IDP data at Admin Level 2
idp_admin2_df <- get_idp_admin2_data(Operation = "Yemen conflict", CountryName = "Yemen")
head(idp_admin2_df)
```

`get_subscription_key` *Retrieval of an API subscription key from the environment.*

Description

The DTM API subscription key is returned, provided that it is available in the R session as an environment variable. Users will usually need to set the DTM_SUBSCRIPTION_KEY environment variable through a .Renviron file or by calling `set_subscription_key()`.

Usage

```
get_subscription_key()
```

Details

On the other hand, if the TESTTHAT environment variable is true, indicating that unit tests are being run by the package maintainers, then the subscription key is returned through different means.

Value

A string representing a given subscription key for the DTM API.

Examples

```
## Not run:
# Generally, calling set_subscription_key() without the key as an argument is best,
# as the user can then be prompted to input the key without typing it directly
# into the console, making it more secure and less likely to exposed.
set_subscription_key()

## End(Not run)
```

`set_subscription_key` *Set the user's API subscription key in order to make the API calls.*

Description

The API will be stored as an environmental variable named "DTM_API_KEY".

Usage

```
set_subscription_key(..., key = NULL)
```

Arguments

`...`	Any argument here is ignored. It exists solely to discourage typing in the subscription key directly, e.g. `set_subscription_key("mysubscriptionkeyhere")`, since doing so will only result in the user being prompted to type the subscription key in a graphical user interface. Rather, if the user wishes to set the subscription key through the console / by programmatic means, then the user must explicitly pass the argument to the `key` parameter as a named parameter. E.g. `set_subscription_key(key = "mysubscriptionkeyhere")`.
`key`	Either NULL or a string representing the key. NULL is preferable: using it will prompt the user to type the subscription key in a graphical user interface that masks it.

Value

Nothing. Creates / overwrites an environmental variable as a side effect.

Examples

```
## Not run:
# Generally, calling set_subscription_key() without the key as an argument is best,
# as the user can then be prompted to input the key without typing it directly
# into the console, making it more secure and less likely to exposed.
set_subscription_key()

## End(Not run)
```

Index